Field-Programmable-Gate-Array-Based System for Network Operations

For more information contact **J. Steven Blessing**
(925) 422-6680, blessing1@llnl.gov

Many of the problems we confront in providing complex software systems center on the inability to achieve high performance without increasing hardware platform size. Currently, many of the systems provided by the Information Operations Assurance Center depend on multiple servers operating in parallel to achieve adequate performance.

We built a field-programmable-gate-arrays (FPGA)-based, system-on-a-chip, which serves as a prototype high-performance platform for various algorithms for network analysis. The FPGA configuration integrates critical components in a single chip, including custom network processing components and a PowerPC CPU.

Project Goals

Our goal was to create a true embedded platform, contained in a single FPGA, which could achieve near wire-speed processing for various custom network applications, while giving software programmers the flexibility to code in high-level languages.

Our prototype system comprises a number of individual intellectual property (IP) cores integrated in a single Xilinx V2P7 FPGA. These IP core system components included various networking hardware modules for handling packet filtering and physical network access, glue logic for integrating our components with the onboard PowerPC microprocessor, and bus and interconnect logic for placing all of the components on the single FPGA platform. In addition, we needed to write software for proof-of-concept that could identify packets of various protocol types.

Relevance to LLNL Mission

The major advantage to our work is high-performance network processing

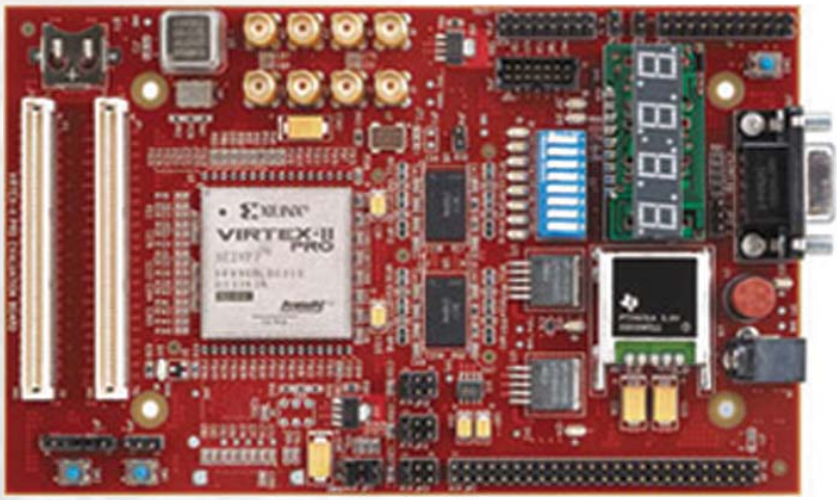

Figure 1. Avnet Virtex 2 Pro board used for prototype.

with a very small system footprint. Such a system could serve as a basis for network analysis applications, or as an autonomous information sensor for distributed network monitoring operations such as firewalls or intrusion detection.

In addition, the individual components could be evolved into production-ready IP cores that are licensed outside of the Laboratory for other uses.

FY2004 Accomplishments and Results

Figures 1 and 2 illustrate our FY2004 effort.

To meet the needs of our packet filtering requirements, we evaluated a number of preliminary architectures. We examined several solutions involving hardware state machines, but eventually decided on a full network microprocessor, due to its flexibility and control.

Silverthrone is an IP core implementation of a network processor built to support the BSD Packet Filter language (BPF). The BPF language is used by many firewall filters for efficient packet classification and filtering. To our knowledge, however, it has only been implemented in software as a virtual machine, and never in actual hardware.

Silverthrone consists of a custom hardware microprocessor that fully implements the BPF instruction set with some additional extensions for packet manipulation. The system executes all BPF instructions in, at most, 3 clock cycles, and mirrors the BPF with a register-based packet filter that uses a directed acyclic control graph. This gives a significant performance increase over a standard expression tree mode, which may contain redundant operations. Our un-optimized prototype implementation of the processor yields clock speeds of approximately 90 MHz, which allows us to identify IP packets in 10 clock cycles. This allows Silverthrone to achieve a theoretical processing rate of 9 million packets per second, given a network front-end capable of writing data at this speed.

With a custom network processor, we also enable users to work with the existing BPF instruction set, allowing the direct use of our architecture without having to understand the underlying hardware. We have also preserved full compatibility with the existing software tools using the BPF instructions.

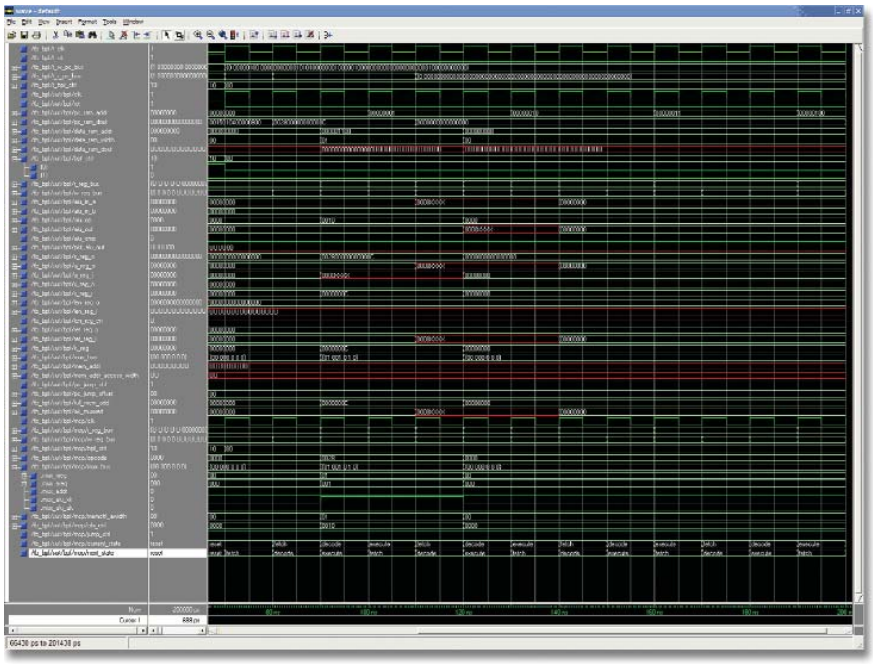

Figure 2. Signal capture of Silverthrone executing single packet instruction.

Silverthrone also contains extensions to the BPF instruction set to allow the network processor to perform more advanced packet analysis than is possible in traditional software based implementations.

Related References

1. Bloom, B., “Space/Time Trade-Offs in Hash Coding with Allowable Errors," *ACM,* **13**, (7), pp. 422-426, May 1970.

2. Dharmapurikar, S., P. Krishnamurthy, T. Sproull, and J. Lockwood, “Deep Packet Inspection Using Parallel Bloom Filters," *Hot Interconnects,* pp. 44-51, August 2003.

3. Ramakrishna, M., E. Fu, and E. Bahcekapili, “A Performance Study of Hashing Functions for Hardware Applications," *Proc. ICCI*, pp. 1621-1636, 1994.

4. Ji, H. M., and M. Carchia, “Fast IP Packet Classification with Configurable Processor," *IEEE,* pp. 2268-2274, 2001.

5. McCanne, S., and V. Jacobson, “The BSD Packet Filter: A New Architecture for User-level Packet Capture," *USENIX Conference Proceedings,* 1993.

FY2005 Proposed Work

We envision extending the existing Silverthrone platform to include analysis-specific operations. These could include: text processing, dynamic session parsing, and stateful packet inspection. The Silverthrone platform can also be extended to provide wireless capabilities such as network mapping and vulnerability assessment.